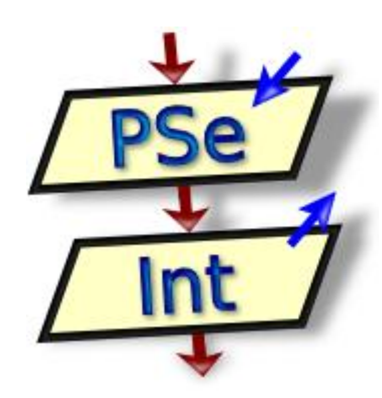
ARREGLOS EN



Nota:

PSeInt es un software importante y muy útil para aquellos estudiantes que se inician en el aprendizaje de la programación. Pero como todo programa informático contiene errores o los famosos *bugs*. Recomiendo como alternativa, mantener el programa actualizado, descargando cada versión nueva que “lancen” de PSeInt, pues la ventaja que obtenemos es la corrección a esos errores y la implementación de mejoras y nuevas funciones.

El autor.

ARREGLOS UNIDIMENSIONALES EN PSEINT

Los arreglos son estructura de datos homogéneas (todos los datos son del mismo tipo) que permiten almacenar un determinado número de datos bajo un mismo identificador, para luego referirse a los mismos utilizando sus posiciones. Los arreglos pueden pensarse como vectores, matrices, etc.

Para crear un arreglo en PSeInt se utiliza la palabra clave **Dimension**, seguido del nombre del arreglo (identificador) y su tamaño entre corchetes [].

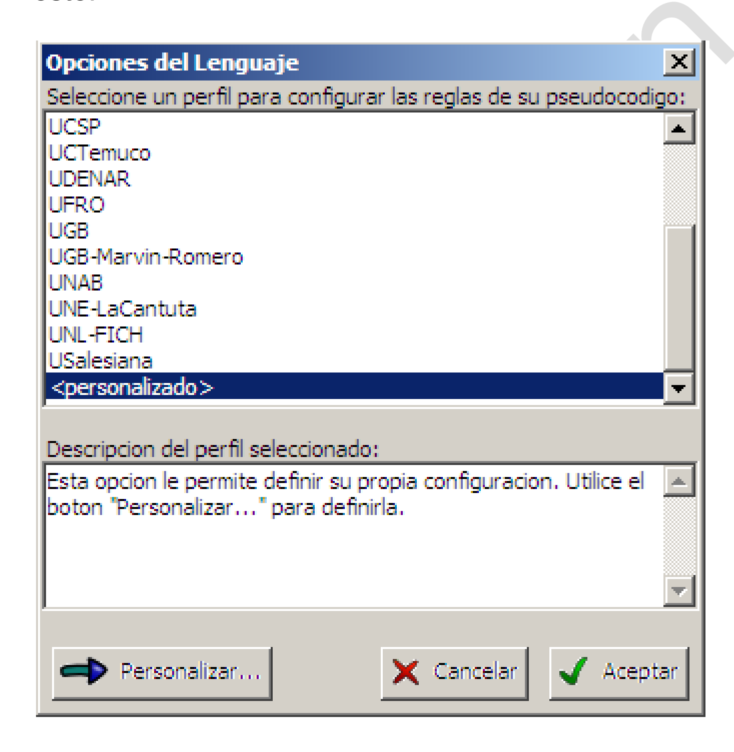
Sintaxis:

Dimension identificador [tamaño];

En PSeInt los arreglos pueden comenzar desde cero o uno. Depende de cómo se configure el programa.

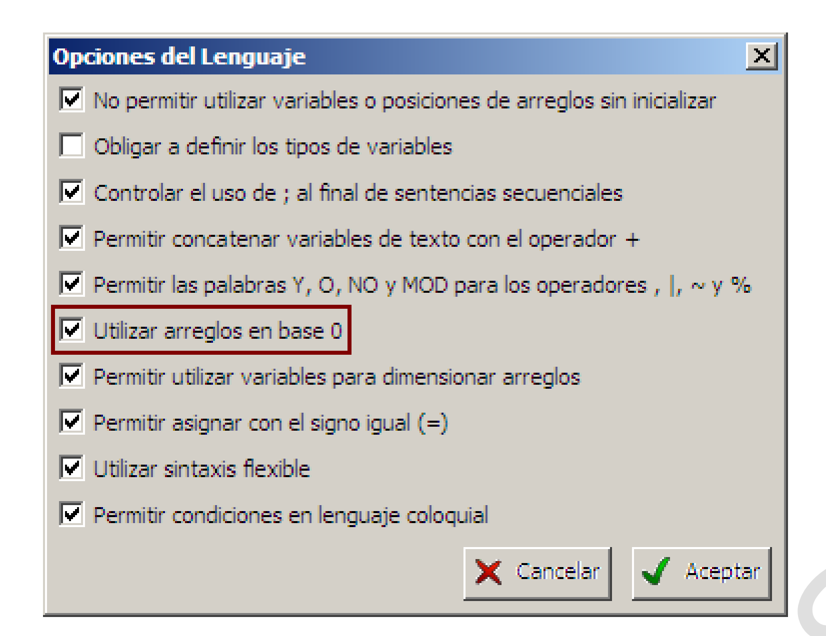
Arreglos en base cero

Para empezar a programar nuestros algoritmos con arreglos en base cero, vamos al menú **configurar** y escogemos **Opciones del lenguaje**, nos saldrá un cuadro como este:



Seleccionamos el perfil **<personalizado>** y hacemos clic en el botón **Personalizar…**

Saldrá el siguiente cuadro:



Marcamos la casilla: *Utilizar arreglos en base cero* y presionamos **aceptar**.

De esta manera ya queda configurado el programa para utilizar arreglos en base cero.

Para comprender mejor el concepto de arreglos se realizará una serie de ejemplos.

Ejemplo 1:

Crear un arreglo llamado **num** que almacene los siguientes datos: 20, 14, 8, 0, 5, 19 y 24.

```
Proceso ejemplo1

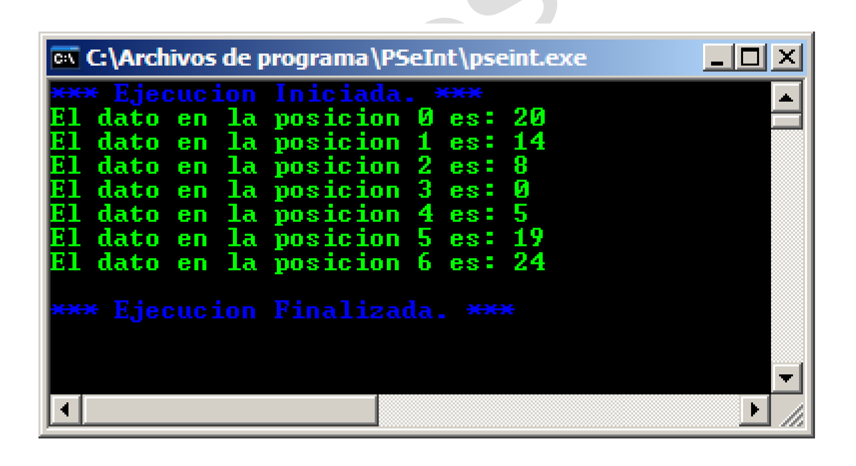
    //creamos el arreglo le damos un nombre y un tamaño de 7 posiciones
    Dimension num[7];

    //a cada posicion le damos un dato
    num[0]<-20;
    num[1]<-14;
    num[2]<-8;
    num[3]<-0;
    num[4]<-5;
    num[5]<-19;
    num[6]<-24;

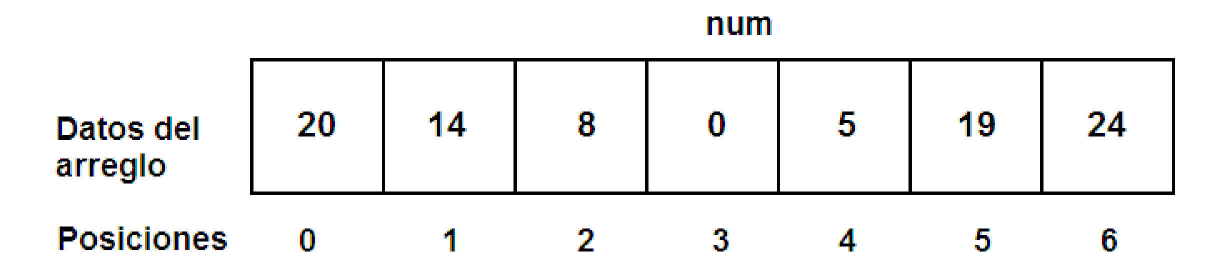
    //imprimimos los datos asignados
    Escribir "El dato en la posicion 0 es: ",num[0];
    Escribir "El dato en la posicion 1 es: ",num[1];
    Escribir "El dato en la posicion 2 es: ",num[2];
    Escribir "El dato en la posicion 3 es: ",num[3];
    Escribir "El dato en la posicion 4 es: ",num[4];
    Escribir "El dato en la posicion 5 es: ",num[5];
    Escribir "El dato en la posicion 6 es: ",num[6];

FinProceso
```

Salida:



Representación grafica del anterior ejemplo:



Al utilizar arreglos en base cero los elementos validos van de **0** a **n-1**, donde n es el tamaño del arreglo. En el ejemplo 1 las posiciones del arreglo **num** entonces van desde 0 a 7-1, es decir de 0 a 6.

Los ciclos, también conocidos como *bucles* o *estructuras de control repetitivas*, juegan un papel muy importante en los arreglos. En el anterior ejemplo, imprimimos los datos a través de siete mensajes, una tarea que lleva cierto tiempo y más cuando la cantidad de datos son demasiados, por eso para facilitar el proceso, utilizamos un ciclo **Para** y así mostrar todos los datos con un sólo mensaje.

Ejemplo 2:

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso ejemplo1

	//creamos el arreglo le damos un nombre y un tamaño de 7 posiciones
	Dimension num[7];

	//a cada posicion le damos un dato
	num[0]<-20;
	num[1]<-14;
	num[2]<-8;
	num[3]<-0;
	num[4]<-5;
	num[5]<-19;
	num[6]<-24;

	//imprimimos los datos asignados con un ciclo para
	Para i<-0 Hasta 7-1 Con Paso 1 Hacer

		Escribir "El dato en la posicion ",i, " es: ",num[i];

	FinPara

FinProceso
```

El ciclo **Para** nos ahorra la tarea de escribir los siete mensajes que muestran los siete datos pedidos inicialmente.

Podemos ver que la salida es la misma:

C:\Archivos de programa\PSeInt\pseint....

```
*** Ejecucion Iniciada. ***
El dato en la posicion 0 es: 20
El dato en la posicion 1 es: 14
El dato en la posicion 2 es: 8
El dato en la posicion 3 es: 0
El dato en la posicion 4 es: 5
El dato en la posicion 5 es: 19
El dato en la posicion 6 es: 24

*** Ejecucion Finalizada. ***
```

Pero no solo podemos imprimir los datos del arreglo con un ciclo, también podemos llenar con datos los arreglos con el ciclo **Para**.

Ejemplo 3:

Crear un arreglo de 5 posiciones y llénelo con los números que el usuario desee.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

    //creamos el arreglo le damos un nombre y un tamaño de 5 posiciones
    Dimension num[5];

    //a cada posicion le damos un dato con el ciclo para
    Para i<-0 Hasta 5-1 Con Paso 1 Hacer

        //pido los datos
        Escribir "Digite un numero para la posicion ",i;
        Leer numero;//capturo los numeros ingresados en la varibale numero

        num[i]<-numero;//al arreglo le paso todos los numeros ingresados

    FinPara

    //imprimimos los datos asignados con un ciclo para
    Para i<-0 Hasta 5-1 Con Paso 1 Hacer

        Escribir "El dato en la posicion ",i, " es: ",num[i];

    FinPara

FinProceso
```

Salida:

C:\Archivos de programa\PSeInt\pseint.exe

```
*** Ejecucion Iniciada. ***
Digite un numero para la posicion 0
> 10
Digite un numero para la posicion 1
> 20
Digite un numero para la posicion 2
> 30
Digite un numero para la posicion 3
> 40
Digite un numero para la posicion 4
> 50
El dato en la posicion 0 es: 10
El dato en la posicion 1 es: 20
El dato en la posicion 2 es: 30
El dato en la posicion 3 es: 40
El dato en la posicion 4 es: 50

*** Ejecucion Finalizada. ***
```

Como se puede apreciar en la salida, los números ingresados por el usuario son: **10**, **20**, **30**, **40**, **50**.

Hemos visto arreglos con datos numéricos, pero también se le pueden llenar con datos de tipo **cadenas de texto**.

Ejemplo 4:

Crear un arreglo de n posiciones y llenarlo con nombres de personas.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

    //pedimos el numero de posiciones o tamaño
    Escribir "Digite el numero de posiciones";
    Leer n;

    //creamos el arreglo le damos un nombre y le pasamos el tamaño ingresado
    Dimension personas[n];

    //a cada posicion le damos un dato con el ciclo para
    Para i<-0 Hasta n-1 Con Paso 1 Hacer

        //pido los datos
        Escribir "Digite un nombre para la posicion ",i;
        Leer nombre;//capturo los nombres ingresados en la varibale nombre

        personas[i]<-nombre;//al arreglo le paso todos los nombres ingresados

    FinPara

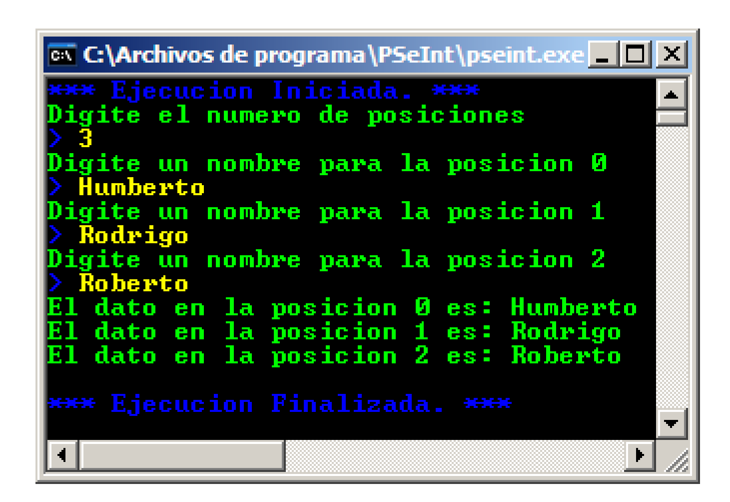
    //imprimimos los datos asignados con un ciclo para
    Para i<-0 Hasta n-1 Con Paso 1 Hacer

        Escribir "El dato en la posicion ",i, " es: ",personas[i];

    FinPara

FinProceso
```

Salida:

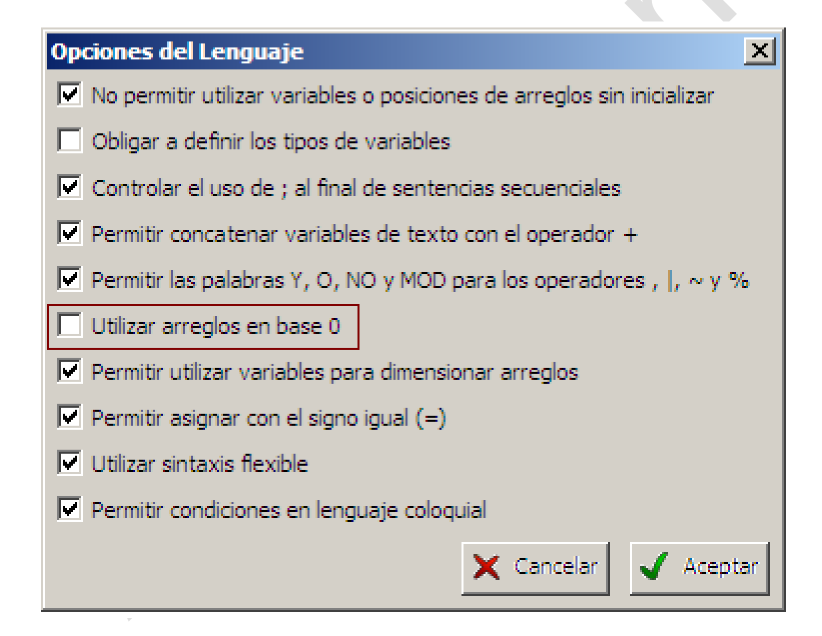


En este ejemplo el usuario eligió 3 posiciones, llenando el arreglo con los siguientes nombres: **Humberto**, **Rodrigo** y **Roberto**.

Arreglos en base 1

Comienzan desde **1** hasta **n**, donde n es el tamaño del arreglo.

Para programar sus algoritmos en base 1 recuerde tener desmarcada la casilla:



Ejemplo 5:

Crear un arreglo de n posiciones y llenarlo con los números que el usuario desee.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

    //pedimos el tamaño
    Escribir "Digite el tamano del arreglo";
    Leer n;

    //creamos el arreglo le damos un nombre y le pasamos el tamaño ingresado
    Dimension num[n];

    //a cada posicion le damos un dato con el ciclo para
    Para i<-1 Hasta n Con Paso 1 Hacer

        //pido los datos
        Escribir "Digite un numero para la posicion ",i;
        Leer numero;//capturo los numeros ingresados en la varibale numero

        num[i]<-numero;//al arreglo le paso todos los numeros ingresados

    FinPara

    //imprimimos los datos asignados con un ciclo para
    Para i<-1 Hasta n Con Paso 1 Hacer

        Escribir "El dato en la posicion ",i, " es: ",num[i];

    FinPara

FinProceso
```

En el ciclo **Para** la variable ya no comienza con cero sino con **uno** y va hasta **n**.

Salida:

C:\Archivos de programa\PSeInt\pseint.exe

```
*** Ejecucion Iniciada. ***
Digite el tamano del arreglo
> 4
Digite un numero para la posicion 1
> 100
Digite un numero para la posicion 2
> 200
Digite un numero para la posicion 3
> 300
Digite un numero para la posicion 4
> 400
El dato en la posicion 1 es: 100
El dato en la posicion 2 es: 200
El dato en la posicion 3 es: 300
El dato en la posicion 4 es: 400

*** Ejecucion Finalizada. ***
```

Consideraciones:

*No importa con que base trabajemos en los arreglos: uno o cero, siempre obtendremos los mismos resultados de forma eficiente, pero recomiendo al estudiante trabajar en PSeInt con arreglos en base cero ya que a la hora de aprender un lenguaje de programación como por ejemplo **Java**, los arreglos siempre van a comenzar desde cero.*

También podemos definir uno o más arreglos en una misma instrucción separándolos con una coma.

Ejemplo 6:

Crear dos arreglos uno que almacene **2** nombres y otro que almacene **3** números.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

	//creamos dos arreglos en una sola instruccion
	Dimension arreglo1[2],arreglo2[3];

	//el arreglo1 almacena datos de cadena
	arreglo1[1]<-"Fernando";
	arreglo1[2]<-"Oscar";

	//el arreglo2 almacena datos numericos
	arreglo2[1]<-74;
	arreglo2[2]<-56;
	arreglo2[3]<-10;

	//imprimimos todos los datos del arreglo 1
	Escribir "Arreglo 1";
	Para i<-1 Hasta 2 Con Paso 1 Hacer

		Escribir "El dato en la posicion ",i," es: ",arreglo1[i];

	FinPara

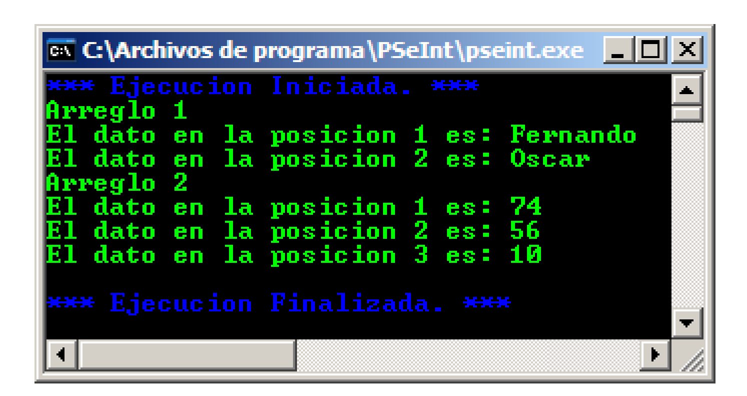
	//imprimimos los datos del arreglo dos
	Escribir "Arreglo 2";
	Para i<-1 Hasta 3 Con Paso 1 Hacer

		Escribir "El dato en la posicion ",i," es: ",arreglo2[i];

	FinPara

FinProceso
```

Salida:



EJERCICIOS RESUELTOS CON ARREGLOS

1. Sumar todos los elementos de un arreglo de tamaño n.

Para sumar los elementos de un vector debemos usar un acumulador inicializado en cero.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

    acum<-0;

    //pedimos el tamaño
    Escribir "Digite el tamano del vector (arreglo)";
    Leer t;

    //creamos el arreglo y le pasamos el tamaño ingresado
    Dimension vector[t];

    //llenamos el arreglo con un ciclo para
    Para i<-1 Hasta t Con Paso 1 Hacer

        //pedimos los numeros
        Escribir "Digite un numero ";
        Leer num;

        vector[i]<-num;//le damos al vector los numeros ingresados
        acum<-acum+vector[i];//acumulamos los numeros y los sumamos

    FinPara

    //imprimimos todos los datos del arreglo
    Para i<-1 Hasta t Con Paso 1 Hacer

        Escribir "La suma de: ",vector[i];

    FinPara

    //resultado total
    Escribir "Es: ",acum;

FinProceso
```

Salida:

```
C:\Archivos de programa\PSeInt\pseint.exe
*** Ejecucion Iniciada. ***
Digite el tamano del vector (arreglo)
> 3
Digite un numero
> 12
Digite un numero
> 1
Digite un numero
> 3
La suma de: 12
La suma de: 1
La suma de: 3
Es: 16

*** Ejecucion Finalizada. ***
```

2. Sumar los elementos de dos vectores y guardar el resultado en otro vector.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

    acum<-0; acum2<-0;//inicializamos los acumuladores en cero

    Escribir "Digite el tamano del vector 1";//pedimos el tamaño para el vector 1
    Leer t;

    Dimension vector1[t];//creamos el primer arreglo y le pasamos el tamaño ingresado

    Para i<-1 Hasta t Con Paso 1 Hacer//llenamos el arreglo con un ciclo para

        Escribir "Digite un numero ",i;//pedimos los numeros
        Leer num;

        vector1[i]<-num;//le damos al vector 1 los numeros ingresados
        acum<-acum+vector1[i];//acumulamos los numeros y los sumamos

    FinPara

    Escribir "Digite el tamano del vector 2";//pedimos el tamaño para el vector 2
    Leer t;

    Dimension vector2[t];//creamos el segundo arreglo y le pasamos el tamaño ingresado

    Para i<-1 Hasta t Con Paso 1 Hacer//llenamos el arreglo con un ciclo para

        Escribir "Digite un numero ",i;//pedimos los numeros
        Leer num;

        vector2[i]<-num;//le damos al vector 2 los numeros ingresados
        acum2<-acum2+vector2[i];//acumulamos los numeros y los sumamos

    FinPara

    Dimension vector3[2];//creamos el arreglo donde guardamos el resultado

    vector3[1]<-acum+acum2;//almacenamos en la posicion 1 del arreglo el resultado

    Escribir "Resultado total <- ",vector3[1];//finalmente mostramos el total

FinProceso
```

Salida:

```
C:\Archivos de programa\PSeInt\...
*** Ejecucion Iniciada. ***
Digite el tamano del vector 1
> 2
Digite un numero 1
> 3
Digite un numero 2
> 4
Digite el tamano del vector 2
> 3
Digite un numero 1
> 10
Digite un numero 2
> 2
Digite un numero 3
> 5
Resultado total <- 24

*** Ejecucion Finalizada. ***
```

3. Llenar un vector de 10 posiciones con números aleatorios entre 1 y 100.

Para los números aleatorios PSeInt utiliza la función **Azar**, ésta escoge un entero aleatorio entre 0 y x-1.

```
PSeInt
Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>

Proceso arreglos

    //creamos el arreglo y le pasamos el tamaño ingresado
    Dimension num[10];

    //llenamos el arreglo con un ciclo para
    Para i<-1 Hasta 10 Con Paso 1 Hacer

        num[i]<-azar(100)+1;//al arreglo le damos numeros aleatorios entre 1 y 100

    FinPara

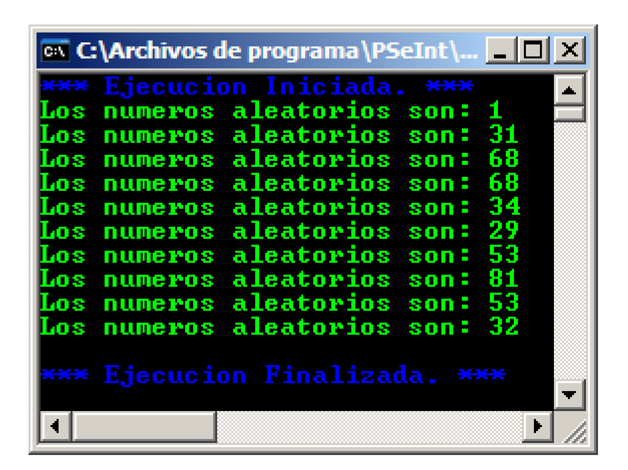
    //imprimimos los datos con un ciclo para
    Para i<-1 Hasta 10 Con Paso 1 Hacer

        Escribir "Los numeros aleatorios son: ",num[i];

    FinPara

FinProceso
```

Salida:



De esta manera cada vez que ejecutemos el algoritmo arrojará distintos números.

4. Llenar un vector con números enteros (números positivos ó negativos). Mostrar la cantidad de números positivos que hay en dicho arreglo.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

    cont<-0;//variable que cuenta los positivos inicializada en cero

    //pedimos el tamano
    Escribir "Digite el numero de posiciones (tamano)";
    Leer p;

    //creamos el arreglo y le pasamos el tamaño ingresado
    Dimension num[p];

    //llenamos el arreglo con un ciclo para
    Para i<-1 Hasta p Con Paso 1 Hacer

        Escribir "Digite numero ",i;//pedimos los numeros
        Leer numero;

        num[i]<-numero;//le pasamos al arreglo los numeros ingresados

    FinPara

    //imprimimos los datos con un ciclo para
    Para i<-1 Hasta p Con Paso 1 Hacer

        Escribir "Los numeros ingresados son: ",num[i];//mostramos los numeros ingresados

        si (num[i]>0) Entonces//si numero ingresado mayor a cero entonces es positivo

            cont<-cont+1;//contamos los numeros que sean positivos

        FinSi

    FinPara

    Escribir "Hay ",cont," numeros positivos";//mostramos cuantos numeros positivos hay

FinProceso
```

Salida:

```
C:\Archivos de programa\PSeInt\pseint.exe
*** Ejecucion Iniciada. ***
Digite el numero de posiciones (tamano)
> 4
Digite numero 1
> -7
Digite numero 2
> 45
Digite numero 3
> 5
Digite numero 4
> -1
Los numeros ingresados son: -7
Los numeros ingresados son: 45
Los numeros ingresados son: 5
Los numeros ingresados son: -1
Hay 2 numeros positivos

*** Ejecucion Finalizada. ***
```


5. Almacene en un arreglo de n posiciones nombres de países. Implementar una opción que al digitar una posición muestre el dato que contiene.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso arreglos

    //pedimos la cantidad
    Escribir "Digite el numero de paises";
    Leer n;

    //creamos el arreglo y le pasamos el tamaño ingresado
    Dimension paises[n];

    //llenamos el arreglo con un ciclo para
    Para i<-1 Hasta n Con Paso 1 Hacer

        Escribir "Digite pais ",i;//pedimos los paises
        Leer nombre;

        paises[i]<-nombre;//le pasamos al arreglo los paises ingresados

    FinPara

    //pedimos la posicion
    Escribir "Digite la posicion";
    Leer pos;

    //mostramos el pais que se encuentra en la posicion digitada
    Escribir "El pais que hay en la posicion ",pos," es: ",paises[pos];

FinProceso
```

Salida:

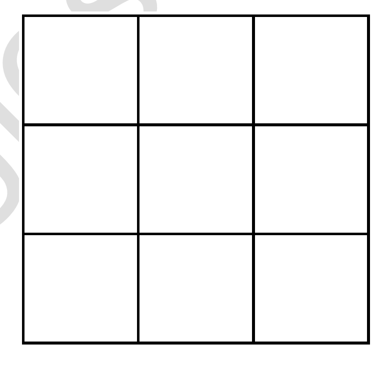
```
C:\Archivos de programa\PSeInt\pseint.exe
*** Ejecucion Iniciada. ***
Digite el numero de paises
> 5
Digite pais 1
> Venezuela
Digite pais 2
> Colombia
Digite pais 3
> Ecuador
Digite pais 4
> Peru
Digite pais 5
> Bolivia
Digite la posicion
> 2
El pais que hay en la posicion 2 es: Colombia

*** Ejecucion Finalizada. ***
```

ARREGLOS BIDIMENSIONALES (MATRICES)

Hasta ahora hemos trabajado con arreglos de una sola dimensión, es decir con un sólo índice, el índice es el número que encerramos dentro de los corchetes (el tamaño del vector).

Un arreglo bidimensional, también conocido como matriz, es parecido a una tabla ya que se compone de n filas y n columnas. Por ejemplo tenemos la siguiente tabla:



Vemos que está compuesta por tres filas y tres columnas. De esta misma forma podemos representar gráficamente a una matriz, como veremos más adelante.

Para crear una matriz en PSeInt se utiliza la palabra clave **Dimension**, seguido del nombre que la identifica y el número de filas y columnas.

Sintaxis:

Dimension identificador [filas,columnas];

Para comprender mejor el concepto de matrices se realizaran algunos ejemplos y ejercicios.

Ejemplo 1:

Crear una matriz **2x2** que almacene los siguientes valores: **10**, **20**, **30**, **40**.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

<sin_titulo>

```
Proceso matriz

	//creamos la matriz y le damos un numero de filas y columnas
	Dimension matriz[2,2];

	//le damos valores a la matriz
	matriz[1,1]<-10;//fila 1 columna 1
	matriz[1,2]<-20;//fila 1 columna 2
	matriz[2,1]<-30;//fila 2 columna 1
	matriz[2,2]<-40;//fila 2 columna 2

	//imprimimos los datos para su visualizacion
	Escribir "El dato que hay en la fila 1 y columna 1 es: ",matriz[1,1];
	Escribir "El dato que hay en la fila 1 y columna 2 es: ",matriz[1,2];
	Escribir "El dato que hay en la fila 2 y columna 1 es: ",matriz[2,1];
	Escribir "El dato que hay en la fila 2 y columna 2 es: ",matriz[2,2];

FinProceso
```

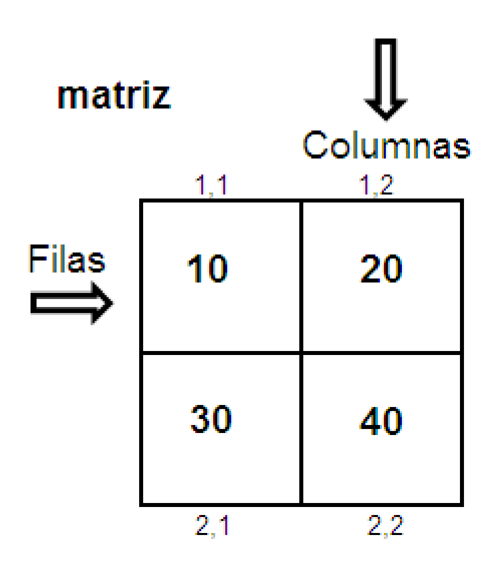
Salida:

C:\Archivos de programa\PSeInt\pseint.exe

```
*** Ejecucion Iniciada. ***
El dato que hay en la fila 1 y columna 1 es: 10
El dato que hay en la fila 1 y columna 2 es: 20
El dato que hay en la fila 2 y columna 1 es: 30
El dato que hay en la fila 2 y columna 2 es: 40

*** Ejecucion Finalizada. ***
```

Representación grafica del anterior ejemplo:



Las filas son **horizontales** y las columnas **verticales**.

En la fila 1 columna 1 el dato es: **10**

En la fila 1 columna 2 el dato es: **20**

En la fila 2 columna 1 el dato es: **30**

En la fila 2 columna 2 el dato es: **40**

Así como en los arreglos unidimensionales llenábamos el vector con un ciclo **Para**, en las matrices también lo podemos hacer, sólo que ya no se utilizará un ciclo sino **dos**, uno para las filas y otro para las columnas. También los datos se muestran con dos ciclos.

Ejemplo 2:

Crear una matriz de n filas y n columnas. Llenar la matriz con los números que el usuario desee.

```
Proceso arreglos

    //pedimos el numero de filas
    Escribir "Digite el numero de filas";
    Leer f;

    //pedimos el numero de columnas
    Escribir "Digite el numero de columnas";
    Leer c;

    //creamos la matriz y le pasamos el numero de filas y columnas ingresados
    Dimension matriz[f,c];

    //llenamos la matriz con dos ciclos PARA, uno para las filas y otro para las columnas
    Para i<-1 Hasta f Con Paso 1 Hacer
        Para j<-1 Hasta c Con Paso 1 Hacer

            //pedimos los datos
            Escribir "Digite dato para la fila ",i," columna ",j;
            Leer numero;

            //llenamos la matriz con los numeros ingresados
            matriz[i,j]<-numero;

        FinPara

    FinPara

    //mostramos todos los datos que hay en la matriz con dos cilos PARA
    Para i<-1 Hasta f Con Paso 1 Hacer
        Para j<-1 Hasta c Con Paso 1 Hacer


            Escribir "Los datos quehay en la matriz son ",matriz[i,j];

        FinPara

    FinPara

FinProceso
```

Salida:

```
C:\Archivos de programa\PSeInt\pseint.exe
*** Ejecucion Iniciada. ***
Digite el numero de filas
> 2
Digite el numero de columnas
> 3
Digite dato para la fila 1 columna 1
> 1
Digite dato para la fila 1 columna 2
> 2
Digite dato para la fila 1 columna 3
> 3
Digite dato para la fila 2 columna 1
> 4
Digite dato para la fila 2 columna 2
> 5
Digite dato para la fila 2 columna 3
> 6
Los datos que hay en la matriz son 1
Los datos que hay en la matriz son 2
Los datos que hay en la matriz son 3
Los datos que hay en la matriz son 4
Los datos que hay en la matriz son 5
Los datos que hay en la matriz son 6

*** Ejecucion Finalizada. ***
```

EJERCICIO CON MATRIZ

1. Crear una matriz n x n y llenarla con los números que el usuario desee. Sume todos los números que componga la columna 1.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

matriz.psc

```
Proceso arreglos

    acum<-0;//acumulador inicializado en cero para sumar los elementos de la columna 1

    Escribir "Digite el numero de filas";//pedimos el numero de filas
    Leer f;

    Escribir "Digite el numero de columnas";//pedimos el numero de columnas
    Leer c;

    Dimension matriz[f,c];//creamos la matriz y le pasamos el numero de filas y columnas ingresados

    //llenamos la matriz con dos ciclos PARA, uno para las filas y otro para las columnas
    Para i<-1 Hasta f Con Paso 1 Hacer
        Para j<-1 Hasta c Con Paso 1 Hacer

            Escribir "Digite dato para la fila ",i," columna ",j;//pedimos los datos
            Leer numero;

            matriz[i,j]<-numero;//llenamos la matriz con los numeros ingresados

        FinPara

        acum<-acum+matriz[i,1];//acumulamos y sumamos todos los numeros que componen la columna 1

    FinPara

    //mostramos todos los datos que hay en la matriz con dos ciclos PARA
    Para i<-1 Hasta f Con Paso 1 Hacer
        Para j<-1 Hasta c Con Paso 1 Hacer

            Escribir matriz[i,j];

        FinPara

    FinPara

    Escribir "Todos los elementos de la columna 1 suman un total de: ",acum;//mostramos la suma

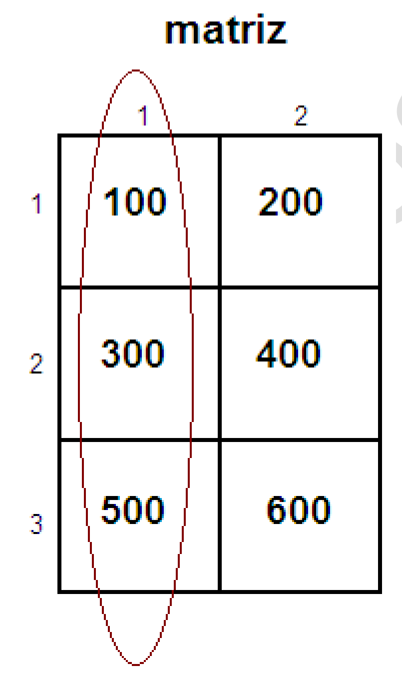
FinProceso
```

Salida:

```
C:\Archivos de programa\PSeInt\pseint.exe
*** Ejecucion Iniciada. ***
Digite el numero de filas
> 3
Digite el numero de columnas
> 2
Digite dato para la fila 1 columna 1
> 100
Digite dato para la fila 1 columna 2
> 200
Digite dato para la fila 2 columna 1
> 300
Digite dato para la fila 2 columna 2
> 400
Digite dato para la fila 3 columna 1
> 500
Digite dato para la fila 3 columna 2
> 600
100
200
300
400
500
600
Todos los elementos de la columna 1 suman un total de: 900

*** Ejecucion Finalizada. ***
```

Representación grafica del anterior ejercicio:



Total: 900

2. Llenar una matriz de 3 x 3 completamente de números aleatorios entre 0 y 9.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

matriz.psc

```
Proceso arreglos

    //creamos la matriz y le pasamos 3 filas y 3 columnas
    Dimension matriz[3,3];

    //llenamos la matriz con dos ciclos PARA, uno para las filas y otro para las columnas
    Para i<-1 Hasta 3 Con Paso 1 Hacer
        Para j<-1 Hasta 3 Con Paso 1 Hacer

            //a la matriz le damos numeros aleatorios entre 0 y 9
            matriz[i,j]<-azar(10);

        FinPara

    FinPara

    //mostramos todos los datos que hay en la matriz
    Para i<-1 Hasta 3 Con Paso 1 Hacer

        Escribir matriz[i,1]," ",matriz[i,2]," ",matriz[i,3];

    FinPara

FinProceso
```

Salida:

C:\Archivos de programa\PSeInt...

```
*** Ejecucion Iniciada. ***
3 1 0
1 9 2
4 6 6

*** Ejecucion Finalizada. ***
```

FIN!

Tutorial realizado los días 15 y 16 de enero de 2012 con la versión **20111106** (6/nov/2011) de **PSeInt**